Make a Radio Telescope With Raspberry Pi

By ElectricSpace in CircuitsRaspberry Pi 33,877 182 27 Featured

(cc) BY-NC-SA

Download Favorite

By **ElectricSpace**

Follow

About: I am a hobbyist who likes space and electronics. More About ElectricSpace »

It is really easy to get an optical telescope. You can just buy one from a manufacturer of such telescopes. However, the same cannot really be said of radio telescopes. Usually, you have to make them yourself. In this Instructable, I

will show how to build a radio telescope that scans the sky within the frequencies of 10.2 GHz and 12.75 GHz.

Add Tip Ask Question Comment Download

Teacher Notes

Teachers! Did you use this instructable in your classroom?
Add a Teacher Note to share how you incorporated it into your lesson.

Add Teacher Note

Step 1: Getting the Parts

0 1 2 3 4 5 6 7 8 9 10
dB
SATELLITE
FINDER

In order to make this radio telescope, first, you will need to get the parts for it.

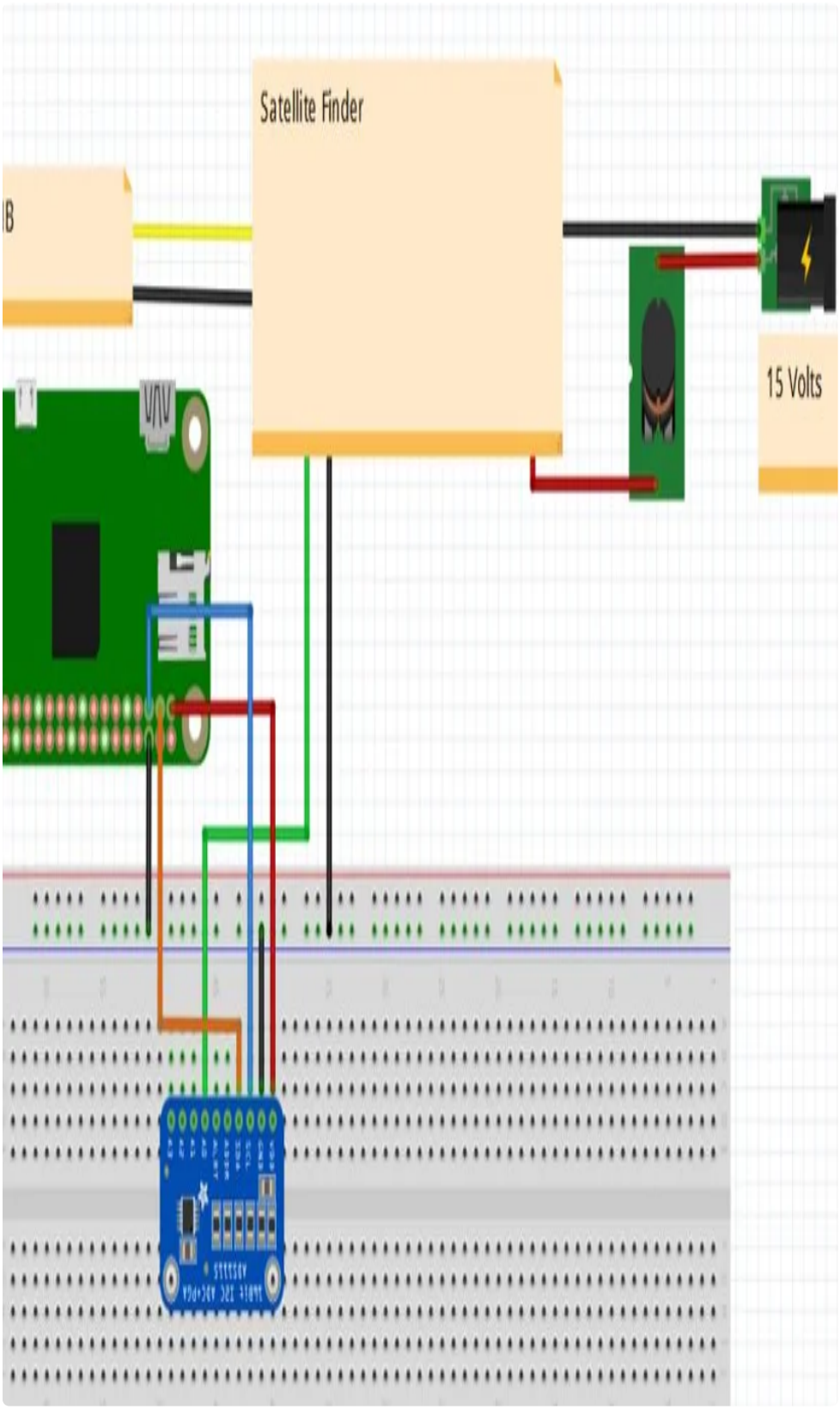
- Satellite Dish with only one LNB mount (can be obtained online, like this, or elsewhere)
- Nylon or teflon washers
- LNB
- Breadboard
- Analog Satellite Finder
- DC Barrel Jack and fitting AC-DC adapter (15 volts for this finder)
- Raspberry Pi with standard peripherals, and an SD card of at least 16GB
- Jumper wires
- 16-bit ADS1115 Analog to Digital Converter
- 100 µH microhenry RF choke
- Hookup wire (I used 22-Guage)
- F-type Coaxial cable of at least 6 feet
- Standard soldering materials

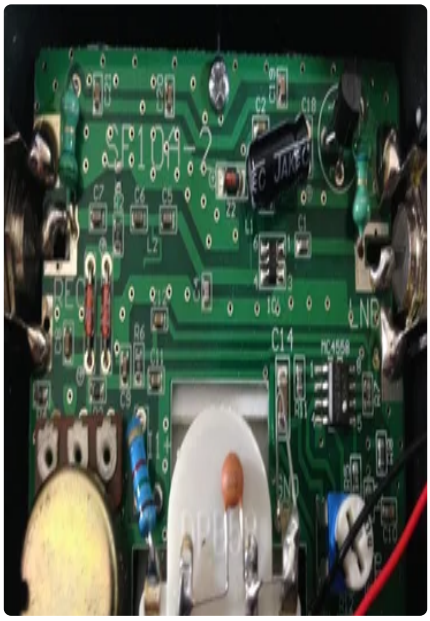
You will also need appropriate software in order to use the radio telescope. You need to have downloaded onto the Raspberry Pi Raspbian, which should include Python 3, and the Python library for the ADS1115.

For your smartphone, you will want to use a satellite tracking app in order to discern between satellites and stellar objects, and a star tracking app to know where celestial objects are in the sky.

Add Tip Ask Question Comment Download

Step 2: Hardware





2 More Images

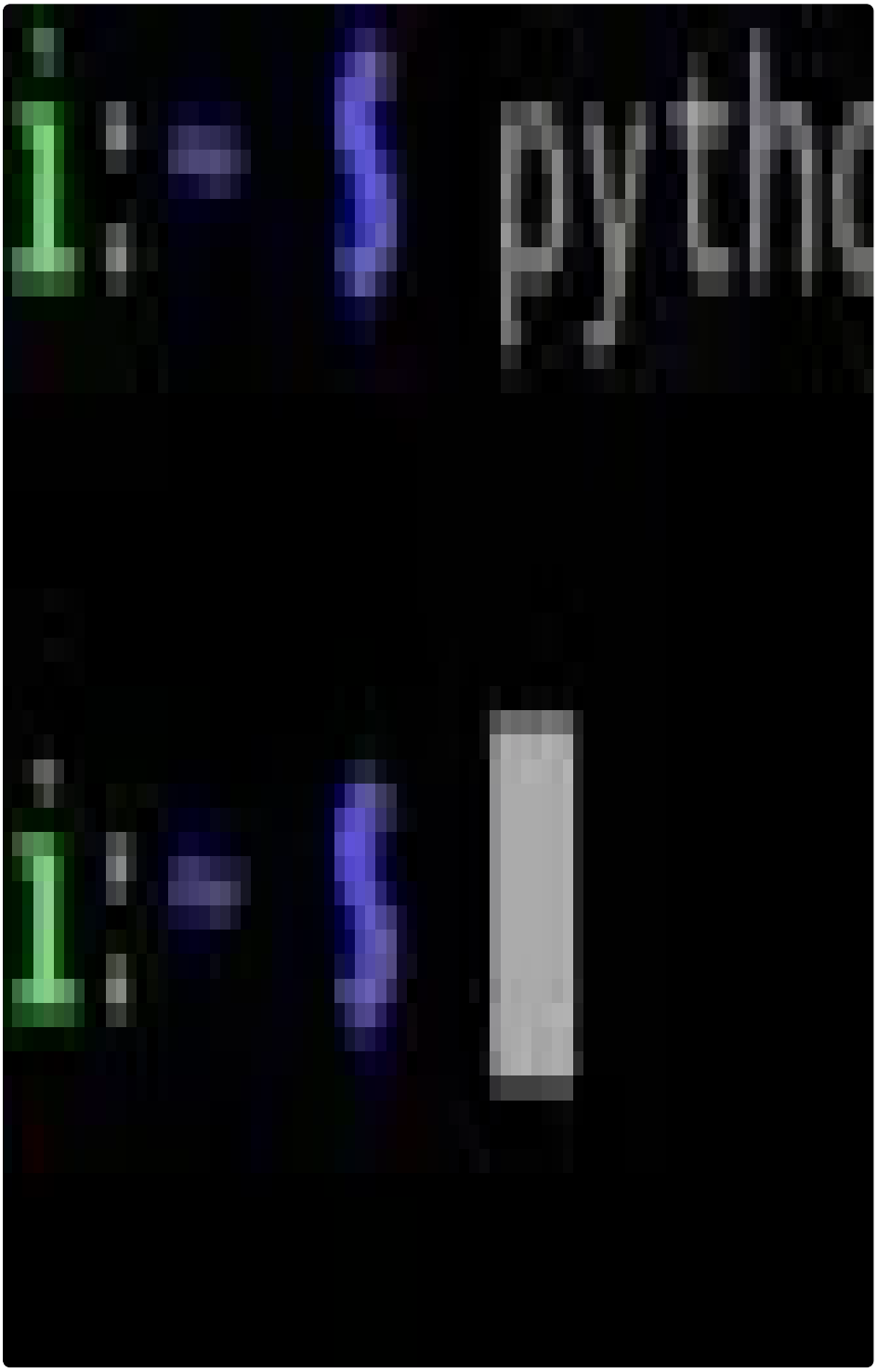
Follow the diagram and pictures shown in making the electronics for the radio telescope.

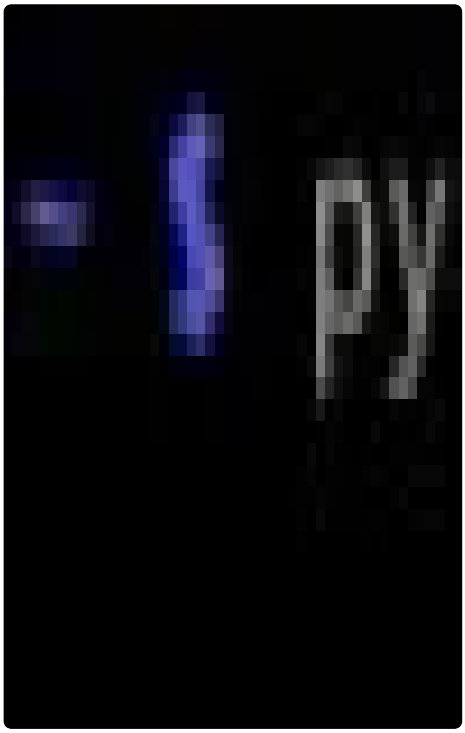
The wires going to the dial of the finder should be disconnected from the dial. The ground connection of the ADS1115 connects to the ground pin that leads to the dial, and the analog input should be connected to the other wire.

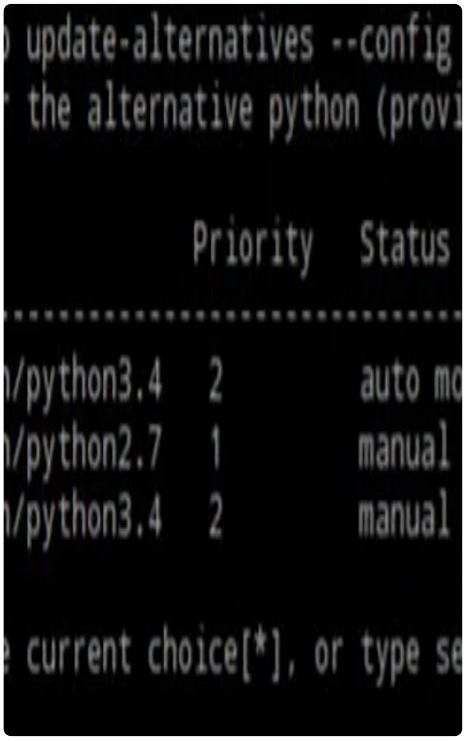
On the dish itself, a nylon washer should be placed between the nut and the backup support.

Add Tip Ask Question Comment Download

Step 3: Software







In order to read and store the data, the Raspberry Pi and the ADS1115 come into play. Any Raspberry Pi with the latest version of Raspbian can do. The instructions for the software library are in the PDF on the Adafruit website. Before downloading, you must set Python 3 as the default Python. In order to check, type into terminal

```
python --version
```

If you get a response that reads Python 3.x.x, the default Python version is Python 3, and you do not need to change the default Python version. However, if your default version is version 2, you will have to change it by going into terminal and typing in

```
sudo update-alternatives --config python
```

Then, press 0 to select Python 3 as the default version. Once you have downloaded the Python library, you can download the code for using the radio telescope. On the Raspberry Pi, create a folder in /home/pi called radio_telescope_files. You should of course have standard peripherals for a Raspberry Pi, such as keyboard, mouse, and monitor. If you have the Raspberry Pi Zero without GPIO pins, you will have to solder them yourself. You will also have to solder the pins on the ADS1115 breakout board.

column.py Download

toScreen.py Download

writeToFile.py Download

Add Tip Ask Question Comment Download

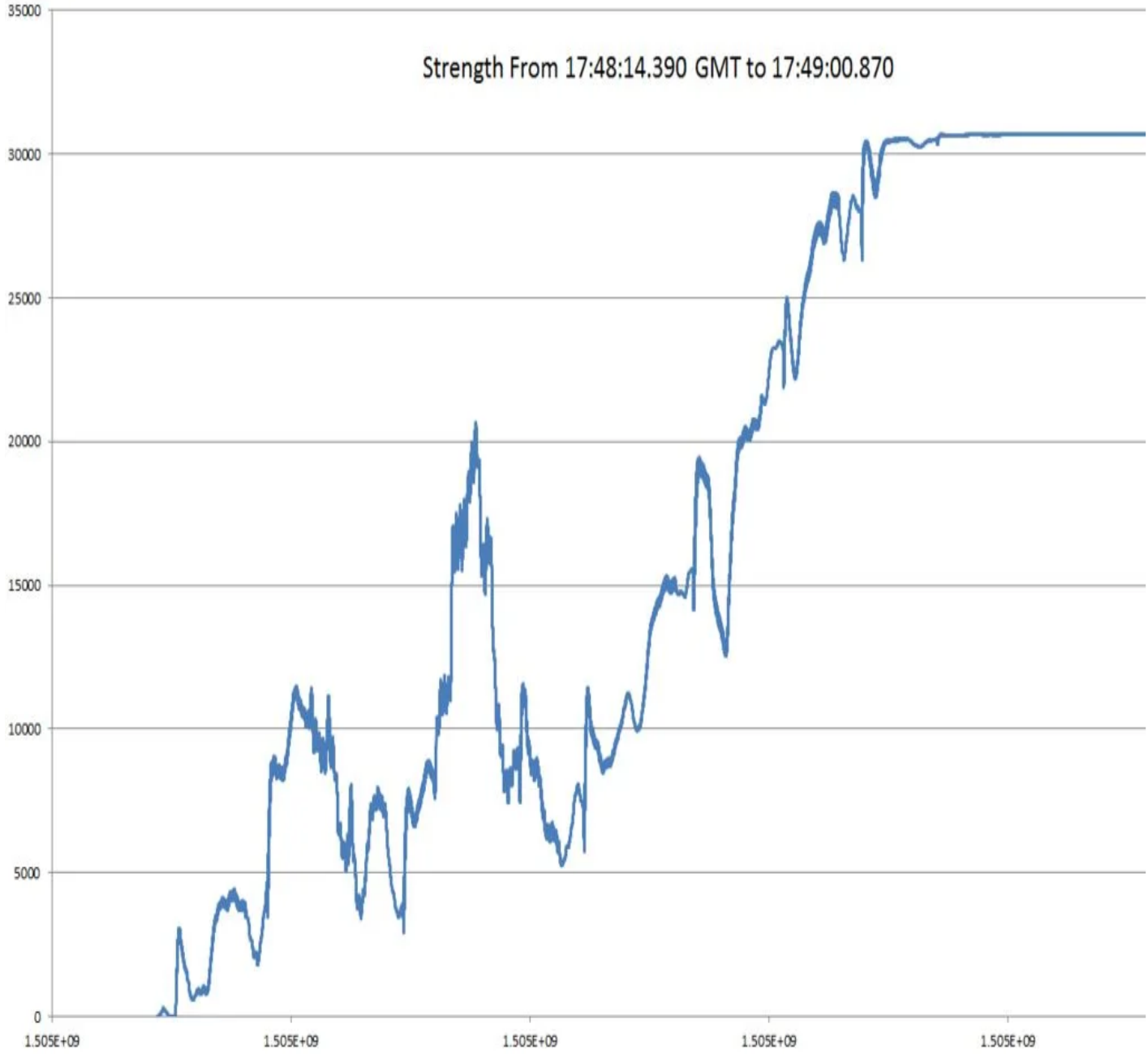
Step 4: Short Tests

Once you have the appropriate software on the Pi, and all the pins soldered on, you can connect the breakout board to the Raspberry Pi. In order to do so, put the pins of the board into a breadboard. The VDD pin should be connected to a 3.3-volt or 5-volt pin on the Raspberry Pi, GND to any ground pin on the Pi, SCL to pin 5 on the Pi, which is SCL, and SDA to pin 3, or SDA, on the Pi. Once the ADS1115 is connected to the Pi, you can now connect the green wire of the modified Finder to A0 on the ADS1115, and the black wire to GND on the board. If it suits you better, you can connect the respective wires by attaching an alligator clip wire to the wire, and a jumper wire to the other end, connecting to the respective board connection. Then, connect the LNB to the input on the Finder by coaxial cable. Plug the power cable into the barrel jack to turn the finder on.

In order to test the Radio telescope, point the dish at such as the sun, the strongest emitter of radio waves from our perspective on Earth. In order to do so, point the dish towards the sun so that the top of the shadow of the LNB hits where the LNB arm meets the dish. Now, turn your Raspberry Pi on and run toScreen.py, the Python script for reading results from the ADS1115 and printing them on screen. You can run this in either the Python 3 IDLE, or terminal. Either way, you should get a prompt asking for the gain, followed by the sample rate, and how long you want the Pi to read the ADS1115's output. With your dish pointed at the sun, run the script for about 10 seconds. If very low numbers initially shows up, turn the gain knob on the Finder up, very slowly. The numbers should increase until it reaches about 30700. By then, you can stop turning the knob.

Add Tip Ask Question Comment Download

Step 5: Saving Results



toScreen.py is a good way to test the radio telescope, but it does not store data. writeToFile.py can store the data, and you can run this in the same way in IDLE and terminal. This script stores data in a text file, which should be found in the folder named ‘Data.’ If you run this, it will ask for the gain, the sample rate, what duration of time you want the Pi to read the ADC, and the name of the file in which you store this data. The radio telescope will pick up the radio signal strength at points throughout the time the radio telescope has been scanning the sky will be stored in the Raspberry Pi.

After collecting the data, it can be graphed in a spreadsheet program, by first getting the timestamps of the data, putting them in column A, then getting the data, and putting them in column B. This can be achieved by using the column.py script. To get the timestamps, run the script, then enter time for the message asking which to read, the timestamps or the data values. In reading the graph, it is important to know that the leftmost point on it represents the westernmost point in the sky that was scanned